DevOps Shack

Dockle

[Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps]

Dockle is a powerful container image linter crafted to elevate the security and adherence to best practices of Docker images. Acting as a linter, Dockle meticulously assesses container images, scrutinizing them for potential security vulnerabilities, ensuring compliance with industry best practices, and enhancing overall image quality. **Key Features:**

1. **Security Assessment:**
    - Dockle performs a meticulous analysis of container images, focusing on security aspects. It adeptly identifies potential vulnerabilities, insecure configurations, and other elements that may expose the image to security risks.
2. **Best-Practice Compliance:**
    - The tool systematically checks Docker images against industry best practices and standards. It guarantees that images adhere to recommended guidelines for configuration, image layering, and other critical factors essential for maintaining a secure and efficient containerized environment.
3. **Ease of Use:**
    - Designed with user-friendliness in mind, Dockle seamlessly integrates into existing workflows. Its intuitive command-line interface empowers users to effortlessly incorporate image linting into their Docker image build and deployment processes.
4. **Practical Recommendations:**

    - Dockle goes beyond identification by providing actionable recommendations based on its analysis. This includes valuable suggestions to enhance image security, reduce vulnerabilities, and optimize configurations in alignment with Docker best practices.

In summary, Dockle stands as a vital tool in the Docker ecosystem, offering robust security assessments, enforcing best practices, and providing practical insights to ensure the creation of Docker images that not only meet but exceed industry standards for security and efficiency.

Checkpoint Summary:

- Details of each checkpoint see CHECKPOINT.md
- CIS's Docker Image Checkpoints

CODE	DESCRIPTION	LEVEL
CIS-DI-0001	Create a user for the container	WARN
CIS-DI-0002	Use trusted base images for containers	FATAL
CIS-DI-0003	Do not install unnecessary packages	FATAL
...	...	...

- Dockle Checkpoints for Docker

CODE	DESCRIPTION	LEVEL
DKL-DI-0001	Avoid sudo command	FATAL
DKL-DI-0002	Avoid sensitive directory mounting	FATAL
DKL-DI-0003	Avoid apt-get dist-upgrade	WARN
...	...	...

- Dockle Checkpoints for Linux

CODE	DESCRIPTION	LEVEL
DKL-LI-0001	Avoid empty password	FATAL
DKL-LI-0002	Be unique UID/GROUPs	FATAL
DKL-LI-0003	Only put necessary files	INFO

Level Definitions:

- Dockle has 5 check levels.

LEVEL	DESCRIPTION
FATAL	Be practical and prudent
WARN	Be practical and prudent, but limited uses (even if official images)
INFO	May negatively inhibit the utility or performance
SKIP	Not found target files
PASS	Not found any problems

Dockle Installation

The provided script is a set of commands for downloading and installing Dockle, a container image linter, on a Linux system. Let's break down the script step by step:

1. **Retrieve the Latest Version from GitHub Releases:**

```
VERSION=$(
  curl --silent "https://api.github.com/repos/goodwithtech/dockle/releases/latest" | \
  grep '"tag_name":' | \
  sed -E 's/.*"v([^"]+)".*/\1/' \
)
```

   - This part uses `curl` to query the GitHub API for the latest release information of Dockle.

- The response is then processed using `grep` and `sed` to extract the version number of the latest release.

6. **Download the Dockle Debian Package:**

```
curl -L -o dockle.deb
https://github.com/goodwithtech/dockle/releases/download/v${VERSION}/
dockle_${VERSION}_Linux-64bit.deb
```

   - This command uses `curl` to download the Debian package (`.deb`) for the specified version of Dockle.

7. **Install the Dockle Debian Package:**

```
sudo dpkg -i dockle.deb && rm dockle.deb
```

   - This command installs the downloaded Debian package using `dpkg`.
   - `&&` ensures that the second command (`rm dockle.deb`) is executed only if the installation is successful.
   - The second command removes the downloaded Debian package after installation to clean up the system.

In summary, this script fetches the latest version of Dockle from GitHub releases, downloads the corresponding Debian package, installs it using `dpkg`, and then removes the downloaded package. This is a common approach for installing software on Debian-based Linux systems. Ensure that you have the necessary permissions (sudo) to install packages on your system before running this script.
It looks like you've provided a series of commands related to Dockle, a container image linter for security. Let's break down each command:

1. **Basic Dockle Scan:**

```
dockle [YOUR_IMAGE_NAME]
```

   - This command runs Dockle to scan the Docker image specified by `[YOUR_IMAGE_NAME]`. It checks for security issues and adherence to best practices.

2. **Dockle Scan with JSON Output:**

```
dockle -f json -o results.json IMAGE_NAME
```

   - This command runs Dockle on the specified Docker image (`IMAGE_NAME`) and outputs the results in JSON format to a file named `results.json`. The `-f` flag sets the output format, and `-o` specifies the output file.

3. **Set Non-Zero Exit Code on Warnings or Errors:**

```
dockle --exit-code 1 [IMAGE_NAME]
```

- By default, Dockle exits with code 0 even if there are problems. This command uses the `--exit-code` option to exit with a non-zero exit code if WARN or FATAL alerts are found during the scan.

4. **Specify Specific Checks to Ignore:**

```
dockle -i CIS-DI-0001 -i DKL-DI-0006 [IMAGE_NAME]
```

   - This command runs Dockle on the specified Docker image and ignores the checks with IDs CIS-DI-0001 and DKL-DI-0006. The `-i` option is used to specify checks to ignore.

5. **Docker Hub Authentication:**

```
export DOCKLE_AUTH_URL=https://registry.hub.docker.com
export DOCKLE_USERNAME={DOCKERHUB_USERNAME}
export DOCKLE_PASSWORD={DOCKERHUB_PASSWORD}
```

   - These commands set environment variables for Docker Hub authentication. The `DOCKLE_AUTH_URL` variable specifies the Docker registry URL, and `DOCKLE_USERNAME` and `DOCKLE_PASSWORD` are set to your Docker Hub username and password.

Please note that storing sensitive information such as passwords in plaintext environment variables may pose security risks. Consider using Docker credentials store or other secure methods for handling credentials, especially in production environments.

These commands demonstrate how to use Dockle for container image security scanning, customize output formats, control exit codes based on findings, and ignore specific checks during the scan.